

THUẬT TOÁN QUAY LUI (BACKTRACKING)

1. Phương pháp (thuật toán quay lui Backtracking)

Có ba dạng cơ bản trong các thuật toán quay lui:

Dạng 1: Tìm tất cả các nghiệm.

Dạng 2: Tìm một nghiệm.

Dạng 3: Tìm nghiệm tối ưu thỏa mãn điều kiện.

2. Dạng 1: Tìm tất cả các nghiệm

Mô hình của thuật toán quay lui có thể mô tả như sau:

```
void Try( int i) {Thủ tục này thử cho  $x_i$  nhận lần lượt các giá trị mà nó có thể nhận}
{
  for (<mọi giá trị j có thể gán cho  $x_i$ >)
  {
    <Thử cho  $x_i = j$ >;
    if (< $x_i$  là phần tử cuối cùng trong cấu hình>)
      <Thông báo cấu hình tìm được> ;
    else
    {
      <Ghi nhận việc cho  $x_i$  nhận giá trị j (Nếu cần)>;
      Try(i + 1);           {Gọi đệ quy để chọn tiếp  $x_{i+1}$ }
      <Nếu cần, bỏ ghi nhận việc thử  $x_i = j$ , để thử giá trị khác>;
    }
  }
};
```

Thuật toán quay lui thường bắt đầu bằng lời gọi Try(1)

3. Dạng 2 : Tìm một nghiệm

```
void Try( int i) {Thủ tục này thử cho  $x_i$  nhận lần lượt các giá trị mà nó có thể nhận}
{
  for (<mọi giá trị j có thể gán cho  $x_i$ >)
  {
    <Thử cho  $x_i = j$ >;
    if (< $x_i$  là phần tử cuối cùng trong cấu hình>)
      { <Thông báo cấu hình tìm được> ; exit(); }
    else
    {
      <Ghi nhận việc cho  $x_i$  nhận giá trị j (Nếu cần)>;
      Try(i + 1);           {Gọi đệ quy để chọn tiếp  $x_{i+1}$ }
      <Nếu cần, bỏ ghi nhận việc thử  $x_i = j$ , để thử giá trị khác>;
    }
  }
};
```

VÍ DỤ 1. LIỆT KÊ DÃY NHỊ PHÂN ĐỘ DÀI N

Input: file văn bản NHIPHAN.INP chứa số nguyên dương N ($N \leq 30$).

Output: file văn bản NHIPHAN.OUT ghi các dãy nhị phân, mỗi dãy trên một dòng.

NHIPHAN.INP	NHIPHAN.OUT
4	0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111

VÍ DỤ 2. SƠN CÔNG NGHIỆP

Một nhà máy sản xuất sơn công nghiệp sản xuất ra N loại sơn khác nhau (mỗi loại sơn có một màu khác nhau). Từ N màu sơn đó nhà máy muốn tạo ra số lượng màu sơn khác đa dạng hơn bằng cách đưa trộn lẫn các loại sơn này lại với nhau để cho ra màu sơn mới. Bạn hãy giúp nhà máy xem từ N loại sơn, mỗi cách trộn là đưa trộn K loại sơn khác nhau lại với nhau. Hỏi có tất cả bao nhiêu cách trộn. Các loại sơn được được đánh số lần lượt là 1, 2, 3, ..., N.

Input: file văn bản SON.INP chứa hai số nguyên dương N, K ($1 \leq K \leq N \leq 30$) cách nhau ít nhất một dấu cách.

Output: file văn bản SON.OUT ghi các cách trộn trong N loại sơn {1, 2, ..., n}, mỗi cách trên một dòng, dòng cuối cùng ghi tổng số cách trộn,

SON.INP	SON.OUT
---------	---------

5 3	{1,2,3} {1,2,4} {1,2,5} {1,3,4} {1,3,5} {1,4,5} {2,3,4} {2,3,5} {2,4,5} {3,4,5}
-----	------------------------------------------------------------------------------------------------------------

VÍ DỤ 3. LÁT GẠCH

Để lát gạch trang trí lối đi cho một trường mầm non, nhà thầu quyết định mua N loại gạch cùng kích thước (mỗi viên gạch có kích thước là 1×1) nhưng có màu khác nhau để lát. Trên lối đi được chia thành các ô hình chữ nhật có kích thước là $1 \times K$ ($1 \leq K \leq N$). Hỏi có bao nhiêu cách lát trên mỗi ô, thứ tự các loại gạch được lát khác nhau trong một ô được xem là một cách lát khác nhau. Các loại gạch được đánh số từ $1, 2, 3, \dots, N$.

Input: file văn bản GACH.INP chứa hai số nguyên dương N, K ($1 \leq K \leq N \leq 20$) cách nhau ít nhất một dấu cách.

Output: file văn bản GACH.OUT ghi các cách lát từ N loại gạch $\{1, 2, \dots, N\}$, mỗi cách gồm K loại gạch.

GACH.INP	GACH.OUT
3 2	1 2 1 3 2 1 2 3 3 1 3 2

VÍ DỤ 4. BÀI TOÁN PHÂN TÍCH SỐ

Hãy tìm tất cả các cách phân tích số n thành tổng của các số nguyên dương, các cách phân tích là hoán vị của nhau chỉ tính là 1 cách.

Input: file văn bản PHANTICH.INP chứa số nguyên dương n ($n \leq 30$).

Output: file văn bản PHANTICH.OUT ghi các cách Ý tưởng số n .

PHANTICH.INP	PHANTICH.OUT
5	5 = 1+1+1+1+1 5 = 1+1+1+2 5 = 1+1+3 5 = 1+2+2 5 = 1+4 5 = 2+3 5 = 5

VÍ DỤ 5. TRÒ CHƠI TRÊN BẢNG GIẤY (Hội thi GV Giỏi Hà Tĩnh 2012)

Trên một bảng giấy người ta kẻ N ô vuông liên tiếp nhau theo chiều ngang. Các ô vuông được đánh số từ 1 đến N theo chiều từ trái sang phải ($1 < N < 50$). Tại ô vuông số 1 có đặt một quân cờ. Một nước đi trong trò chơi là dịch chuyển quân cờ về phía bên phải 1 hoặc 2 ô vuông.

Hãy đếm và liệt kê tất cả các cách dịch chuyển quân cờ từ ô vuông số 1 tới ô vuông N (Ký hiệu: ký hiệu 'O' mô tả dịch chuyển quân cờ về phía bên phải 1 ô vuông, ký hiệu 'T' mô tả dịch chuyển quân cờ về phía bên phải 2 ô vuông).

Dữ liệu vào là tệp văn bản BANG.INP ghi số nguyên dương N .

Dữ liệu ra là tệp văn bản BANG.OUT có cấu trúc:

- Dòng thứ nhất ghi số nguyên M là số cách dịch chuyển quân cờ từ ô vuông 1 đến ô vuông N .
- M dòng tiếp theo mỗi dòng ghi một cách dịch chuyển trong số M cách di chuyển quân cờ từ ô số 1 tới ô số N .

*			
---	--	--	--

1 2 3 4

BANG.INP	BANG.OUT
4	3 OOO OT TO

VÍ DỤ 6. BÀI TOÁN XẾP N QUÂN HẬU

Xét bàn cờ tổng quát kích thước $n \times n$. Một quân hậu trên bàn cờ có thể ăn được các quân khác nằm tại các ô cùng hàng, cùng cột hoặc cùng đường chéo. Hãy tìm cách xếp n quân hậu trên bàn cờ sao cho không quân nào ăn quân nào.

Input: file văn bản QUEENS.INP chứa số nguyên dương $n \leq 12$.

Output: file văn bản QUEENS.OUT, mỗi dòng ghi một cách đặt n quân hậu.

QUEENS.INP	QUEENS.OUT
5	(1, 1); (2, 3); (3, 5); (4, 2); (5, 4); (1, 1); (2, 4); (3, 2); (4, 5); (5, 3); (1, 2); (2, 4); (3, 1); (4, 3); (5, 5); (1, 2); (2, 5); (3, 3); (4, 1); (5, 4); (1, 3); (2, 1); (3, 4); (4, 2); (5, 5); (1, 3); (2, 5); (3, 2); (4, 4); (5, 1); (1, 4); (2, 1); (3, 3); (4, 5); (5, 2); (1, 4); (2, 2); (3, 5); (4, 3); (5, 1); (1, 5); (2, 2); (3, 4); (4, 1); (5, 3); (1, 5); (2, 3); (3, 1); (4, 4); (5, 2);

Ví dụ một cách xếp với $n = 5$

VÍ DỤ 7. XỔ SỐ ĐIỆN TOÁN

Có N người (đánh số từ 1 đến N) tham gia một đợt xổ số điện toán. Mỗi người nhận được một thẻ gồm M ô (đánh số từ 1 đến M). Người chơi được chọn K ô trong số các ô đã cho bằng cách đánh dấu các ô được chọn. Sau đó các thẻ này được đưa vào máy tính để xử lý. Máy tính chọn ra K ô ngẫu nhiên (gọi là các ô kết quả) và chấm điểm từng thẻ dựa vào kết quả đã sinh. Cứ mỗi ô chọn đúng với ô kết quả thì thẻ chơi được tính 1 điểm. Giả thiết biết các ô chọn cũng như các điểm tương ứng của từng thẻ chơi, hãy xác định tất cả các kết quả có thể có mà máy sinh ra.

INPUT: Dữ liệu vào đọc từ file văn bản
XOSO.INP gồm:

Dòng đầu ghi các số N, M, K.

Dòng thứ i trong N dòng tiếp ghi thẻ
người i gồm K+1 số: K số đầu là các số
các ô chọn, cuối cùng là điểm tương

5 9 4	1 2 3 4
2 4 6 8 2	2 3 4 7
5 6 8 9 0	
2 4 5 6 2	
1 2 3 7 3	
3 5 6 9 1	

-
-
chơi của
hiệu của
ứng.

OUTPUT: Ghi kết quả ra file văn bản

XOSO.OUT, mỗi dòng là một kết quả gồm K số ghi số hiệu các ô mà máy đã sinh.

Ghi chú:

- Các số trên cùng một dòng trong các file vào/ ra, được ghi cách nhau ít nhất một dấu cách trắng.
- Giới hạn kích thước: $N \leq 100$, $M \leq 50$, $K \leq 10$.
- Dữ liệu vào trong các test là hợp lệ và đảm bảo có ít nhất một đáp án.

Ví dụ:

3. Dạng 2 : Tìm một nghiệm

Procedure Try(i: Integer);{Thủ tục này thử cho x_i nhận lần lượt các giá trị mà nó có thể nhận}

Begin

for <mọi giá trị j có thể gán cho x_i > **do**

Begin

<Thử cho $x_i := j$ >;

If < x_i là phần tử cuối cùng trong cấu hình> **then**

Begin

<Thông báo cấu hình tìm được>

Halt; {thoát khỏi chương trình}

End

Else

Begin

<Ghi nhận việc cho x_i nhận giá trị j (Nếu cần)>;

Try(i + 1); {Gọi đệ quy để chọn tiếp x_{i+1} }

<Nếu cần, bỏ ghi nhận việc thử $x_i := j$, để thử giá trị khác>;

End;

End;

End;

VÍ DỤ 1. BÀI TOÁN MÁY RÚT TIỀN TỰ ĐỘNG ATM 1

Một máy ATM hiện có n ($n \leq 20$) tờ tiền có mệnh giá $t_1, t_2, t_3, \dots, t_n$. Hãy đưa ra cách trả tiền đúng bằng s.

Input: Tập ATM.INP có dạng:

- Dòng đầu là hai số n và s.
- Dòng thứ hai gồm n số $t_1, t_2, t_3, \dots, t_n$

Output: Tập ATM.OUT có dạng: Nếu có thể trả đúng s thì đưa ra cách trả, nếu không ghi -1.

ATM.INP	ATM.OUT
11 490	200 20 20 50 100 100
20 200 10 20 20 50 50 50 50 100 100	

4. Dạng 3. Tìm nghiệm tối ưu thỏa mãn điều kiện

Dựa trên mô hình thuật toán quay lui, ta xây dựng mô hình sau:

Procedure khoitao;

Begin

<Khởi tạo một cấu hình bất kỳ CAUHINH>;
End;

{Thủ tục này thử chọn cho x_i tất cả các giá trị nó có thể nhận}

Procedure Try(i: Integer);

Begin

for <Mọi giá trị j có thể gán cho x_i >**do**

Begin

<Thử cho $x_i := j$ >;

If <Việc thử trên vẫn còn hi vọng tìm ra cấu hình tốt hơn CAUHINH> **then**

If < x_i là phần tử cuối cùng trong cấu hình> **then**

<cập nhật CAUHINH>

Else

Begin

<Ghi nhận việc thử $x_i = j$ (nếu cần)>;

Try(i + 1); {Gọi đệ quy, chọn tiếp x_{i+1} }

<Bỏ ghi nhận việc thử cho $x_i = j$ (nếu cần)>;

End;

End;

End;

Begin

khoitao;

Try(1);

<Thông báo cấu hình tối ưu CAUHINH>;

End.

VÍ DỤ 1. BÀI TOÁN MÁY RÚT TIỀN TỰ ĐỘNG ATM 2

Một máy ATM hiện có n ($n \leq 20$) tờ tiền có mệnh giá $t_1, t_2, t_3, \dots, t_n$. Hãy tìm cách trả ít tờ nhất với số tiền đúng bằng s .

Input: Tập ATM.INP có dạng:

- Dòng đầu là hai số n và s .
- Dòng thứ hai gồm n số $t_1, t_2, t_3, \dots, t_n$

Output: Tập ATM.OUT có dạng: Nếu có thể trả đúng s thì đưa ra số tờ ít nhất cần trả và đưa ra cách trả, nếu không ghi -1.

ATM.INP	ATM.OUT
11 390	5
20 200 10 20 20 50 50 50 50 100	200 20 20 50 100
100	

VÍ DỤ 2. BÀI TOÁN NGƯỜI DU LỊCH

Cho n thành phố đánh số từ 1 đến n và m tuyến đường giao thông hai chiều giữa chúng, mạng lưới giao thông này được cho bởi bảng C cấp $n \times n$, ở đây $C_{ij} = C_{ji}$ = Chi phí đi đoạn đường trực tiếp từ thành phố i đến thành phố j . Giả thiết rằng $C_{ii} = 0$ với $\forall i$, $C_{ij} = +\infty$ nếu không có đường trực tiếp từ thành phố i đến thành phố j .

Một người du lịch xuất phát từ thành phố 1, muốn đi thăm tất cả các thành phố còn lại mỗi thành phố đúng 1 lần và cuối cùng quay lại thành phố 1. Hãy chỉ ra cho người đó hành trình với chi phí ít nhất.

Input: file văn bản TOURISM.INP

- Dòng 1: Chứa số thành phố n ($1 \leq n \leq 20$) và số tuyến đường m trong mạng lưới giao thông.
- m dòng tiếp theo, mỗi dòng ghi số hiệu hai thành phố có đường đi trực tiếp và chi phí đi trên quãng đường đó (chi phí này là số nguyên dương ≤ 100).

Output: file văn bản TOURISM.OUT

- Dòng đầu tiên ghi hành trình.
- Dòng thứ 2 ghi chi phí tìm được.

TOURISM.INP	TOURISM.OUT
4 6 1 2 3 1 3 2 1 4 1 2 3 1 2 4 2 3 4 4	1->3->2->4->1 6

2292. Trò chơi dò mìn

Mã bài: NKMINES

Một bãi mìn hình chữ nhật có cạnh $M \times N$ nguyên dương. Bãi mìn được chia thành $M \times N$ ô vuông đơn vị bằng các đường song song với các cạnh, các dòng ô vuông đánh số từ 1 đến M từ trên xuống dưới, các cột ô vuông đánh số từ 1 đến N từ trái sang phải, hai ô vuông khác nhau được gọi là kề nhau nếu chúng có ít nhất một đỉnh chung. Mỗi ô vuông có không quá một quả mìn. Để ghi nhận tình trạng mìn tại các ô đồng thời có thể giữ bí mật phần nào, người ta lập một mảng hai chiều M dòng N cột mà $A[U, V]$ bằng số ô mìn có điểm chung với ô $[U, V]$ của bãi mìn (có nhiều nhất 8 ô có điểm chung với một ô cho trước).

Cho mảng A , hãy tìm cách xác định các ô có mìn.

Dữ liệu

- Dòng đầu gồm hai số nguyên M, N là kích thước hình chữ nhật.
- M dòng sau, mỗi dòng ghi N số thể hiện mảng A .

Kết quả

Gồm M dòng, mỗi dòng ghi N số 0 hoặc 1 tương ứng với ô đó không có mìn hoặc có mìn. Nếu có nhiều kết quả thỏa mãn, chỉ cần đưa ra một kết quả duy nhất. Biết rằng dữ liệu vào luôn đảm bảo có ít nhất một kết quả.

Giới hạn

- $1 \leq M, N \leq 200$

Ví dụ

Dữ liệu:

```
4 4
1 3 3 1
2 3 4 4
3 6 5 3
1 3 3 3
```

Kết quả

```
1 0 0 1
0 1 1 0
0 0 1 1
1 1 1 0
```